

What is claimed is:

1. A method for acquiring access to an object in a multi-threaded, object-based system, the method comprising:
 - associating a monitor with the object;
 - 5 setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object; and
 - setting a reference substantially directly from the thread to the monitor.
- 10 2. A method for acquiring access to an object as recited in claim 1 wherein setting the pointer and setting the reference occur substantially when the thread acquires a lock to the object.
- 15 3. A method for acquiring access to an object as recited in claim 1 wherein setting the reference includes setting a boolean reference count.
- 20 4. A method for acquiring access to an object as recited in claim 1 wherein setting the reference enables a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented by the garbage collector.
- 25 5. A method for acquiring access to an object as recited in claim 1 wherein associating a monitor with the object includes obtaining the monitor from a freelist or monitors.
6. A method for acquiring access to an object as recited in claim 1 wherein associating the monitor with the object includes:
 - setting a pointer from the monitor to the object; and
 - setting a pointer from the object to the monitor.
- 30 7. A method for reducing overhead associated with providing a monitor for an object included in a multi-threaded, object-based computing system, the computing

system including a plurality of monitors which includes the first monitor, the method comprising:

- 5 setting a pointer from the first monitor to the object;
- setting a pointer from the object to the first monitor;
- 10 setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object; and
- setting a reference substantially directly from the thread to the first monitor, wherein setting the reference include updating contents of a reference field associated with the thread.

10

- 8. A method for reducing overhead as recited in claim 7 further including:
 - invoking the object using the thread, wherein invoking the object using the thread includes initiating a wait action, the wait action being arranged to place the object into a wait state.

15

- 9. A method for reducing overhead as recited in claim 8 further including:
 - performing a memory reclamation during the wait state.

20

- 10. A method for reducing overhead as recited in claim 9 wherein performing a memory reclamation during the wait state includes:
 - identifying the contents of the reference field;
 - using the contents of the reference field to identify the first monitor; and
 - updating the contents of a monitor field associated with the first monitor to indicate that the first monitor is in use.

25

- 11. A method for reducing overhead as recited in claim 10 wherein performing a memory reclamation during the wait state further includes:
 - scanning through substantially all monitors included in the plurality of monitors; and

30

- reclaiming substantially any monitor included in the plurality of monitors that is not indicated as being in use, wherein the first monitor is not reclaimed because the

contents of the monitor field associated with the first monitor indicate that the first monitor is in use.

12. A computer program product for acquiring access to an object in a multi-threaded, object-based system, the computer program product comprising:
5 computer code for associating a monitor with the object;
computer code for setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object;
computer code for setting a reference substantially directly from the thread to
10 the monitor; and
a computer-readable medium that stores the computer codes.

13. A computer program product according to claim 12 wherein the computer code for setting the reference includes computer code for setting a boolean reference
15 count.

14. A computer program product according to claim 12 wherein the computer code for setting the reference enables a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented
20 by the garbage collector.

15. A computer program product according to claim 12 wherein the computer-readable medium is one selected from the group consisting of a hard disk, a floppy disk, a data signal embodied in a carrier wave, a tape drive, an optical drive, and a
25 CD-ROM.

16. A computer program product for reducing overhead associated with providing a monitor for an object included in a multi-threaded, object-based computing system, the computing system including a plurality of monitors which includes the first
30 monitor, the computer program product comprising:
computer code for setting a pointer from the first monitor to the object;
computer code for setting a pointer from the object to the first monitor;

computer code for setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object;

computer code for setting a reference substantially directly from the thread to the first monitor, wherein setting the reference include updating contents of a reference field associated with the thread; and

5 a computer-readable medium that stores the computer codes.

18. A computer program product according to claim 17 further including:

computer code for invoking the object using the thread, wherein invoking the 10 object using the thread includes initiating a wait action, the wait action being arranged to place the object into a wait state; and

computer code for performing a memory reclamation during the wait state.

19. A computer program product according to claim 18 wherein the computer 15 code for performing a memory reclamation during the wait state includes:

computer code for identifying the contents of the reference field;

computer code for using the contents of the reference field to identify the first monitor;

computer code for updating the contents of a monitor field associated with the 20 first monitor to indicate that the first monitor is in use

computer code for scanning through substantially all monitors included in the plurality of monitors; and

computer code for reclaiming substantially any monitor included in the plurality of monitors that is not indicated as being in use, wherein the first monitor is 25 not reclaimed because the contents of the monitor field associated with the first monitor indicate that the first monitor is in use.

20. A computer program product according to claim 16 wherein the computer-readable medium is one selected from the group consisting of a hard disk, a floppy 30 disk, a data signal embodied in a carrier wave, a tape drive, an optical drive, and a CD-ROM.

21. A multi-threaded, object-based computing system comprising:

- an object;
- a thread, the thread including a reference to the object; and
- a monitor, the monitor including a reference to the object, wherein the object

5 includes a reference to the monitor and the thread includes a reference to the monitor.